

Appl. No. 09/844,345
Amdt. Dated January 20, 2005
Reply to Office action of December 22, 2004

REMARKS/ARGUMENTS

Claims 1-38 are pending in the present application.

This Amendment is in response to the final Office Action mailed December 22, 2004. In the final Office Action, the Examiner objected to the specification, rejected claims 1-38 under 35 U.S.C. §102(e). Reconsideration in light of the remarks made herein is respectfully requested.

Objection to the Specification

In the Final Office Action, the Examiner maintained the objection to the Specification for not having a Summary of the Invention. The Examiner argues that since 37 C.F.R. §1.73 does not state the Examiner should not object to a specification when the brief summary of invention is not present, then the Examiner can object to the specification (Final Office Action, Page 8, item 8). Applicant respectfully disagrees for the following reasons.

The United States Patent Examiners are governed by the statutes and rules as authorized by the 37 C.F.R. The Examiner cannot perform an act or make a decision without proper authorization or support by proper statutes or rules. There are many other things that the 37 C.F.R. does not state the Examiner should not do, but it does not mean that the Examiner is allowed to do those things.

Accordingly, Applicant respectfully requests the objection to the Specification be withdrawn.

Rejection Under 35 U.S.C. § 102

1. In the final Office Action, the Examiner rejected claims 1-38 under 35 U.S.C. §102(e) as being anticipated by U.S. Patent No. 6,077,313 issued to Ruf ("Ruf"). Applicant respectfully traverses the rejection and contends that the Examiner has not met the burden of establishing a prima facie case of anticipation. To anticipate a claim, the reference must teach every element of the claim. "A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." Vergegaal Bros. v. Union Oil Co. of California, 814 F.2d 628, 631, 2 USPQ 2d 1051, 1053 (Fed. Cir. 1987). "The identical invention must be shown in as complete detail as is contained in the...claim." Richardson v. Suzuki Motor Co., 868 F.2d 1226, 1236, 9 USPQ 2d 1913, 1920 (Fed. Cir. 1989).

Appl. No. 09/844,345
Amdt. Dated January 20, 2005
Reply to Office action of December 22, 2004

Applicant reiterates the arguments presented in the previously filed response. Specifically, Applicant contends that Ruf does not disclose, either expressly or inherently, (1) pruning, (2) local graphs representing local problems that correspond to separately compilable components, and (3) values of each of the local graphs form a lattice under a partial ordering.

Ruf merely discloses a type partitioned dataflow analyses. Type partitional dataflow analysis performs a dataflow analysis of a program by partitioning the dataflow analysis into phases (Ruf, col. 6, lines 61-65). The partitioning is performed based on a dependence relation over types representing run-time. A dependence analysis determines the dependence relation among types of the corrupted type relation (Ruf, col. 7, lines 20-22). Dependence analysis represents the dependence relation in the form of a dependence graphs (Ruf, col. 9, lines 39-41). Partitioning algorithm module collapses each strongly-connected component of dependence graph into a single node (Ruf, col. 9, lines 66-67; col. 10, line 1). The resulting dependence graph is a directed acyclic graph (DAG) corresponding to a partial ordering of type representatives (Ruf, col. 10, lines 8-9).

First, Ruf merely discloses transforming a dependence relation by merging depending types in the dependence relation into a single type. Transforming and/or merging is not the same as pruning. Pruning is in essence reducing whereas merging is combining. The two processes are different. Second, Ruf does not disclose local graphs representing local problems corresponding to separately compilable components. Ruf merely discloses a dependence graph. Each type in the dependence relation is represented by a vertex or node in the graph (Ruf, col. 9, lines 42-43). A directed edge represents the dependence between each pair of types (Ruf, col. 9, lines 43-45). Since the vertex/node represents a type, it does not correspond to separately compilable components in a software program. Third, Ruf does not disclose values of the local graph form a lattice under a partial ordering. Ruf merely discloses a directed acyclic graph (DAG) which represents a collapsed dependence graph (Ruf, col. 10, lines 6-9).

In response to the above arguments, the Examiner states that the instant specification recites, "Local inter-procedural problems are constructed for each translation unit, reduced, and merged together into a global problem to be solved (0034 page 8)" (Final Office Action, page 9, item 1). Applicant respectfully disagrees. The cited paragraph merely states that the local problems are reduced and merged together into a global problem. Claim 1 recites "pruning local

Appl. No. 09/844,345
Amdt. Dated January 20, 2005
Reply to Office action of December 22, 2004

graphs representing local problems”, which is the “reduced” phase of the cited paragraph. Claim 1 does not recite the “merged” phase of the cited paragraph.

The Examiner further states that Ruf specifically states that “Partitioning algorithm module . . . may merge suitable type representatives of the ordered dependence relation into a single type representative to help reduce execution time and/or memory space costs in performing the dataflow analysis (col. 10 lines 47-60)”. The Examiner then concludes that “[t]herefore, merging includes reducing in Ruf’s partitioning algorithm, accordingly, it helps ‘reduce peak storage requirements for the dataflow analysis (col. 10 lines 31-47). Applicant respectfully disagrees. Reducing execution time or peak storage requirements is not the same as pruning the local graphs. In fact, Ruf clearly states that this reduction of execution time is achieved by merging the representatives of the ordered dependence relation into a single type representative (Ruf, Col. 10, lines 47-51). Furthermore, type representatives of the ordered dependence relation are not local graphs that represent local problems corresponding to separately compilable software components.

The Examiner further states that a type in Ruf corresponds to a separately compilable component because it also represents a function (Final Office Action, page 9, item 2). Applicant respectfully disagrees. First, Ruf merely states that dataflow analysis dependencies may be determined for types representing functions (Ruf, Col. 3, lines 1-4). Ruf does not disclose that the local graphs represent the local problems corresponding to separately compilable components. Dataflow analyses merely generate a model of every program quantity of interest (Ruf, Col. 1, lines 23-26). They do not prune local graphs. Second, a function is not a separately compilable component. In fact, Ruf’s dataflow analysis is used to approximate the expected run-time behavior of a program (Ruf, Col. 1, lines 15-17). The functions, if present, exist in the same program, not as separately compilable components. A function, by itself, cannot be compiled.

The Examiner further contends that “the applicant simply states that Ruf does not disclose the limitation, values of the local graph form a lattice under a partial ordering, in the claims and fails to show why the limitation is different from the teachings of Ruf.” (Final Office Action, Page 10, item 3). Applicant respectfully disagrees. It is the Examiner’s burden to show that the prior art reference discloses the claimed invention. Here, the Examiner has not met this

Appl. No. 09/844,345
Amdt. Dated January 20, 2005
Reply to Office action of December 22, 2004

burden of showing that Ruf discloses values of the local graphs form a lattice under a partial ordering.

The Examiner further states that Ruf's directed acyclic graph (DAG) corresponds to a partial ordering of type representatives (Ruf, Col. 10, lines 1-10). However, Ruf's DAG does not correspond to values of the local graphs. Ruf's DAG represents a dependence graph. A dependence graph represents dependencies among types using the mapping of program quantities to types as defined by type relation (Ruf, Col. 10, lines 7-10; Col. 9, lines 28-32). In contrast, the local graphs correspond to separately compilable components, which do not contain dependencies among types.

Therefore, Applicant believes that independent claims 1, 15, 29 and their respective dependent claims are distinguishable over the cited prior art references. Accordingly, Applicant respectfully requests the rejection under 35 U.S.C. §102(c) be withdrawn.

Appl. No. 09/844,345
Amdt. Dated January 20, 2005
Reply to Office action of December 22, 2004

Conclusion

Applicant respectfully requests that a timely Notice of Allowance be issued in this case.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

Dated: January 20, 2005

By


Thinh V. Nguyen

Reg. No. 42,034

Tel.: (714) 557-3800 (Pacific Coast)

12400 Wilshire Boulevard, Seventh Floor
Los Angeles, California 90025

CERTIFICATE OF MAILING/TRANSMISSION (37 CFR 1.8A)

I hereby certify that this correspondence is, on the date shown below, being:


MAILING

☐ deposited with the United States Postal Service
as first class mail in an envelope addressed to:
Commissioner for Patents, PO Box 1450,
Alexandria, VA 22313-1450.

FACSIMILE

☒ Transmitted by facsimile to the Patent and
Trademark Office.

Date: January 20, 2005


Tu Nguyen

January 20, 2005

Date